# aula 9 – Introdução ao dom

Breve revisão:

Qual a diferença entre usar =, ==, === em códigos escritos em JS? = serve para adicionar um valor a uma variável por exemplo(recebe), == simbolizam o que o símbolo de apenas um igual simbolizaria numa expressão(igual), === servem para verificar se as restrições de uma string e numero são as mesmas(restrições).

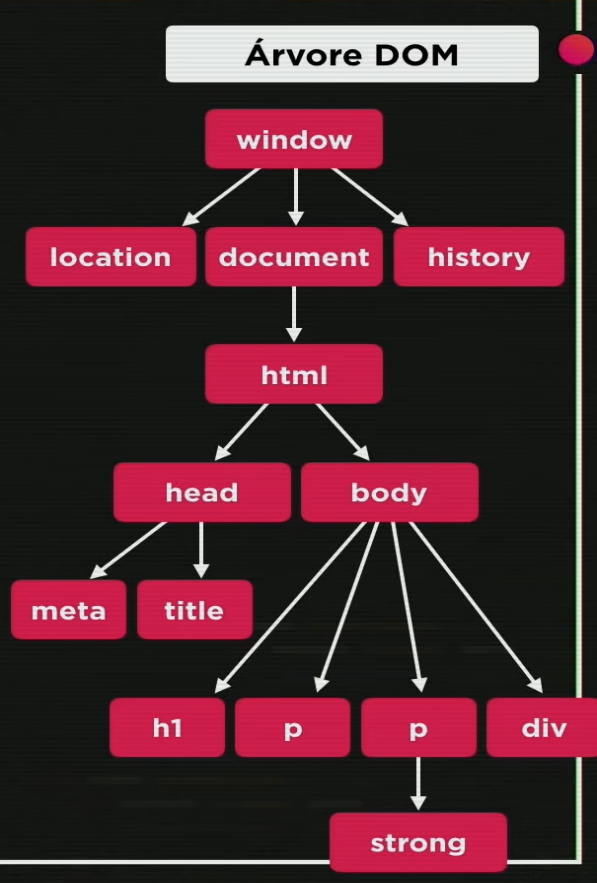
Em uma mesma expressão, temos os operadores || e &&. Quem eu resolvo primeiro? Resolvemos primeiro o &&(conjunção) e depois a ||(disjunção).

Sabe usar o operador ternário para colocar um entre dois calores em uma vaviavel? Precisamos praticar ...

• entendendo o dom

DOM é um acrônimo para Document Object Model(Modelo de objetos para documentos), basicamente é um conjunto de objetos dentro próprio navegador que vai dar acesso aos componentes internos do seu site.

• árvore dom

 Basicamente a arvore dom começa na raiz e essa dentro do navegador a gente chama de window, todo JS do site está dentro de um objeto chamado window, e dentro do window a gente tem vários outros objetos:

Essa arvore tem uma hierarquia, que a gente fala assim, o que está acima na arvore é ‘parent’ (pai/mãe etc), e o que está abaixo a gente chama de ‘child’ (filho). Dentro de window temos muitos outros objetos além dos que tem nessa imagem ao lado, mas por questões de praticidade nesse exemplo usamos apenas alguns para facilitar nossa vida.

• Location: esse objeto diz qual é sua URL, diz qual é pagina anterior, qual a pagina atual etc ...

• Document: é o documento atual;

• History: vai guardar de onde você veio e para onde vai ...

Podemos manipular alguns dados da arvore DOM dentro de um site para executar algumas coisas, abaixo temos alguns exemplos disso:

• Por marca

- getElementsByTagName(): com ele podemos pegar mais de uma tag, exemplo:

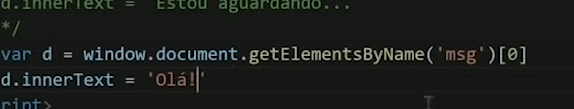
var p1 = window.document.getElementsByTagName('p')[1] // em parênteses e aspas, vai pegar o elemento p(paragrafo), em colchetes vai pegar na ordem que aparece ex:[0] vai pegar o primeiro paragrafo,[1] vai pegar o segundo e assim vai ...

• Por ID

- getElementById(): eu posso identificar um objeto DOM por um seletor id do CSS.

• Por Nome

- getElementsByName(): se eu tenho um atributo ‘name’ em um elemento HTML, posso seleciona-lo deste jeito.



• Por Classe

- getElementByClassName()

• Por Seletor

- querySelector



Funciona tanto com id ( # ) ou com class ( . )

Repare que, sempre que tenho getElemets, no plural, preciso sempre declarar qual objeto quero pegar usando colchetes [].